Received July 16, 2020, accepted July 25, 2020, date of publication August 5, 2020, date of current version August 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3013706

Point Cloud Registration Algorithm Based on the Grey Wolf Optimizer

YAQING FENG[1,2], JIALIN TANG[1,3], BINGHUA SU[1,2], QINGLANG SU[3], AND ZHUANG ZHOU[1,2]

[1]Beijing Institute of Technology, Zhuhai 519088, China
[2]Beijing Institute of Technology, Beijing 100081, China
[3]Research Institute of Data Science, City University of Macau, Macau, China

Corresponding author: Jialin Tang (tang_jl@bitzh.edu.cn)

This work was supported in part by the Key Program of Guangdong University under Grant 2019KZDXM060, and in part by the Innovative Youth Program of Guangdong University under Grant 2019KQNCX194.

ABSTRACT In view of the long computation time and low registration accuracy of the current point cloud registration algorithm, a point cloud registration algorithm based on the grey wolf optimizer (GWO) is proposed, denoted PCR-GW. The algorithm uses the centralization method to solve the translation matrix and then simplifies the points of the initial point cloud models by using the intrinsic shape signatures (ISS) feature. Next, various parameters of the rotation matrix are obtained via the GWO algorithm by employing the quadratic sum of the distances between corresponding points in the simplified point cloud as the objective function. Finally, the point cloud registration process is completed by using the obtained transformation matrix. By conducting a registration experiment on the point cloud library model and comparing PCR-GW with the traditional algorithms, the algorithm proposed in this article is shown to be promising for improving the computation speed and registration accuracy.

INDEX TERMS Point cloud registration, feature point extraction, grey wolf optimization algorithm.

I. INTRODUCTION

In recent years, 3D reconstruction has been extensively applied in medical images, industrial inspection, self-driving cars, cultural relic reconstruction, and indoor modeling [1]. In addition, it has been widely used in aerospace, agriculture, and other fields. The 3D model is built through the steps of data collection, point cloud registration, surface reconstruction, and texture mapping. In the process of data collection, due to the limited visibility of the scanning system, the scanner needs to scan multiple angles and then splice the data to obtain a complete point cloud model. In other words, the point clouds from different angles must be merged into a unified coordinate system, which is known as point cloud registration. As shown in Figure 1, the point cloud in (a) transforms into the point cloud in (b) through rotation and translation and finally merges into a complete point cloud model in the same coordinate system. The result of point cloud registration can directly affect the accuracy of the point cloud model; thus, point cloud registration is a key step in the construction of the point cloud model.

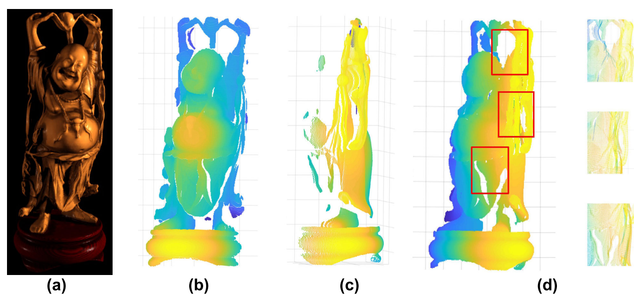

FIGURE 1. Point cloud registration process for the Happy Buddha [2]. (a) The point cloud data scanned from the front of the object. (b) The point cloud data scanned from the side of the object. (c) The point cloud model after registration.

According to the process used, point cloud registration methods can be divided into coarse registration and fine registration [3]. The former type makes the point clouds with overlarge initial distances approach each other, and the latter further refines the result of the former. Existing methods either reduce the accuracy to obtain a higher speed or use more time. It is difficult to balance the time and accuracy, so a better method is needed to solve the problem.

The rest of this article is organized as follows: The second section reviews representative work on point cloud

The associate editor coordinating the review of this manuscript and approving it for publication was Diego Oliva.

registration and briefly introduces the algorithm in the paper. The third section explains the principle of point cloud registration. The fourth section provides a detailed description of the PCR-GW method. The performance of the proposed algorithm is validated by comparison with other registration methods in the fifth section. The sixth section summarizes the PCR-GW algorithm and provides future research directions.

II. RELATED WORK

At present, the iterative closest point (ICP) algorithm proposed by Besl and McKay [4] in 1992 is the most widely used fine registration method; it is based on the optimal matching algorithm of the least-squares method for iteratively solving the rotation and translation matrix until the convergence condition is satisfied. The algorithm has high precision and does not need to extract features. However, its long computational time and need for an inclusion relationship for two-point clouds [5] seriously affects the performance of the ICP algorithm. Hence, many experts and scholars have proposed many approaches to improve the algorithm [6]–[8]. Because of this limitation of the algorithm, ICP and its variants need good initialization to avoid falling into a local minimum.

Coarse registration provides a good initial position from which fine registration can improve registration efficiency. Feature-based registration is a classical method used in coarse registration. This method is divided into the following steps: First, key points are selected for use in simplifying the point cloud models with filtering methods, such as the scale invariant feature transform (SIFT) [9] and 3D Harris [10]. Second, the feature descriptors are used to describe the features of points in the point cloud. The fast point feature histogram (FPFH) [11], heat kernel signature (HKS) [12], and viewpoint feature histogram (VFH) [13] descriptors are exploited to encode the local or global shapes of points. Third, there are many strategies to find corresponding points according to the similarity of point features between the source point cloud and the target point cloud. Cirujeda *et al.* [14] proposed a registration strategy based on the game theory method, and there are also strategies based on the genetic algorithm (GA) [1] and particle swarm optimization (PSO) [15]. Finally, the transformation matrix between the two-point clouds is obtained through a series of corresponding points. This kind of algorithm is easy to understand but takes a long time to run and becomes sensitized to noise.

Additionally, there are also some non-feature-based methods. Aiger *et al.* [16] proposed the 4-point congruent set (4PCS) for registration. This method takes advantage of the constant ratio of segments from four coplanar points in the affine transformation, which has a simple feature and strong antinoise performance. However, it is unsuitable for point clouds with many planes and inconspicuous features. The normal distributions transform (NDT) [17] uses the normal distribution of the points in a point cloud for registration with high speed but low accuracy. Lu *et al.* [18] proposed an end-to-end deep neural network model, which takes considerable time to train but has good registration accuracy.

For point cloud registration, the registration accuracy and running time are two important measurement criteria. The registration accuracy is affected by the initial position, and the registration time is correlated with the initial position and the number of points in the point clouds. To solve the problems with the above methods, a point cloud registration method based on the grey wolf optimizer (GWO) algorithm is proposed, denoted PCR-GW. It utilizes the centralization method to solve the translation matrix. The ISS feature is used to simplify the number of points in the point clouds. The quadratic sum of the distances between corresponding points is considered the fitness function, and the parameters in the transformation matrix are solved by the GWO algorithm. Therefore, a good initial position is provided for fine registration. Comparing the experimental results of this method with those of the 4PCS, FPFH, and PSO algorithms, it is shown that the algorithm in this article has better registration accuracy and faster speed.

III. POINT CLOUD REGISTRATION

The main task of point cloud registration is to splice the point clouds obtained from different angles to obtain a more abundant and complete point cloud model. Suppose the point cloud to be registered is $P = \left\{p_i \,|\, p_i \in R^3, i = 1, 2, 3, \cdots, n\right\}$ and the target point cloud is $Q = \left\{q_j \,|\, q_j \in R^3, j = 1, 2, 3, \cdots, m\right\}$. Point cloud registration estimates the transformation matrices R and T to make the corresponding points in two-point clouds overlap as much as possible [19]; that is, $Q = P*R+T+N_i$. N_i is the vector of the noise, and the equations of R and T are given as follows (1) and (2), as shown at the bottom of the page, where δ, θ, and σ are the angles of rotation around the x, y, and z-axes, respectively.

Due to the existence of partial superpositions and actual error, the solution is carried out in a minimized form [20]; namely, the objective (fitness) function is:

$$F(R, T) = min \sum_{i=1}^{n} ||R * p_i + T - q_i||^2, \tag{3}$$

where q_i is the corresponding point to p_i.

$$R = \begin{bmatrix} cos\theta cos\sigma & -cos\theta sin\sigma & sin\theta \\ cos\delta sin\sigma + sin\delta sin\theta cos\sigma & cos\delta cos\sigma - sin\delta sin\theta sin\sigma & -sin\delta cos\theta \\ sin\delta sin\sigma - cos\delta sin\theta cos\sigma & sin\delta cos\sigma + cos\delta sin\theta sin\sigma & cos\delta cos\theta \end{bmatrix} \tag{1}$$

$$T = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T, \tag{2}$$

IV. PCR-GW METHOD

To obtain the optimal R and T matrices, point cloud registration based on the GWO algorithm is proposed for coarse registration. First, the matrix T is solved by using centralization, and then ISS is employed to obtain key points from the P and Q point clouds. Finally, the GWO algorithm is used to obtain the parameters of the rotation matrix according to the defined fitness function. The specific algorithm flow is shown in Table 1.

A. SOLVING FOR MATRIX T

In the coarse registration algorithm, the matrix T can be solved by centralizing the point clouds. This removes the translation error by moving the centers of mass of two-point clouds to the same place. The T matrix can be formulated as:

$$T = \bar{Q} - \bar{P}, \tag{4}$$

where $\bar{Q}$ and $\bar{P}$ represent the center-of-mass coordinates of point clouds Q and P, respectively. At this point, the point cloud transformation matrix is solved by finding the rotation angles δ, θ, and σ, which makes the R matrix optimal.

B. SELECTING KEY POINTS

In actual conditions, the data contain a large number of noise points. Selecting key points to simplify the point clouds can reduce the computation of the algorithm. Several algorithms can be applied to select the key points, such as the rotational projection statistics (RoPS) feature [12] based on deep learning, the ISS, and the signature of the histogram of orientation (SHOT) feature [10]. For the ISS algorithm, a key point can be determined by calculating the eigenvalue of each point in the neighborhood covariance matrix. The ISS descriptor can not only have a fast calculation speed but also obtain chosen points with high repeatability [1].

First, the K closest neighboring points of p_i are searched in the K-D tree. After that, the neighboring points larger than a specified radius are eliminated by the radius constraint. Finally, the neighboring point set $\{p_j\}$ of p_i is obtained. Then, the covariance matrix of point p_i is produced as follows:

$$cov\,(p_i) = \sum_{|p_i - p_j| < r} w_{ij}\,(p_i - p_j)\,(p_i - p_j)^T \Big/ \sum_{|p_i - p_j| < r} w_{ij}. \tag{5}$$

W_{ij} is a weight parameter that is inversely proportional to the distance from p_j to p_i.

The eigenvalues λ_i^1, λ_i^2, and λ_i^3 of the matrix are obtained from Formula. (5), where the values of λ_i are sorted in descending order. If Formula. (6) can be satisfied, then the point can be preserved. ε_1 and ε_2 are the thresholds.

$$\lambda_i^2 \Big/ \lambda_i^1 \leq \varepsilon_1, \quad \lambda_i^3 \Big/ \lambda_i^2 \leq \varepsilon_2 \tag{6}$$

C. GWO ALGORITHM

The grey wolf optimizer (GWO) algorithm is a swarm intelligence algorithm proposed by Mirjalili *et al.* [21]. The algorithm simulates the collective hunting behavior of grey wolves. When finding prey, they track, encircle, and then attack the prey to obtain food. The wolves are divided into four levels, denoted α, β, γ, and ω, where α is at the highest level and is the leader of the wolves; β wolves rank second only to α and assist α in decision-making; γ wolves obey the commands of α and β; ω wolves are the lowest level, and there are many of them. α, β and γ wolves are all selected from the ω wolves.

1) THE REASON TO CHOOSE GWO

The GWO algorithm is regarded as one of the most promising swarm intelligence algorithms and is used to solve problems in various fields, such as power dispatch problems [22], estimates of landslide susceptibility [23], and forecasts of consumption [24]. With the wide use of the GWO algorithm, there are also many improved algorithms, such as the random walk grey wolf optimizer (RW-GWO) [25] and chaotic grey wolf optimization (CGWO) [26], which greatly promote the development of the GWO algorithm.

Compared with the PSO algorithm, salp swarm algorithm (SSA), artificial bee colony (ABC) algorithm, and other algorithms [27], this method has fewer parameters and faster convergence. Its convergence factor can adjust adaptively to consider global and local search capabilities. We compared the GWO algorithm with the classical PSO algorithm and the latest SSA algorithm. Several optimization functions are selected to evaluate the performance of the algorithms, as shown in Figure 2. The results show that GWO has a better convergence performance and global search capability than the other algorithms.

2) MATHEMATICAL MODEL

To mathematically model the wolves, we consider defining the optimal solution as α and the suboptimal solution and third best solution as β and γ. The other feasible solutions in the solution space are denoted as ω. The model describing how wolves track the prey after discovery is proposed as follows:

$$\vec{D} = \vec{C} \circ \vec{X}_p\,(t) - \vec{X}\,(t) \tag{7}$$

$$\vec{X}\,(t+1) = \vec{X}_p\,(t) - \vec{A} \circ \vec{D}. \tag{8}$$

Formula (7) indicates the distance between an individual and the prey, while Formula (8) represents the update of the individual according to the target position. $\circ$ represents the Hadamard product operation. $\vec{X}_p\,(t)$ represents the position of the prey, $\vec{X}\,(t)$ is the position of an individual wolf, and t is the number of iterations. $\vec{A}$ and $\vec{C}$ are expressed as coefficient vectors:

$$\vec{A} = a(2\vec{r}_1 - 1) \tag{9}$$

$$\vec{C} = 2\vec{r}_2, \tag{10}$$

where a represents the convergence factor, which linearly decreases from 2 to 0 as the number of iterations increases. $\vec{r}_1$ and $\vec{r}_2$ are random vectors in [0, 1].

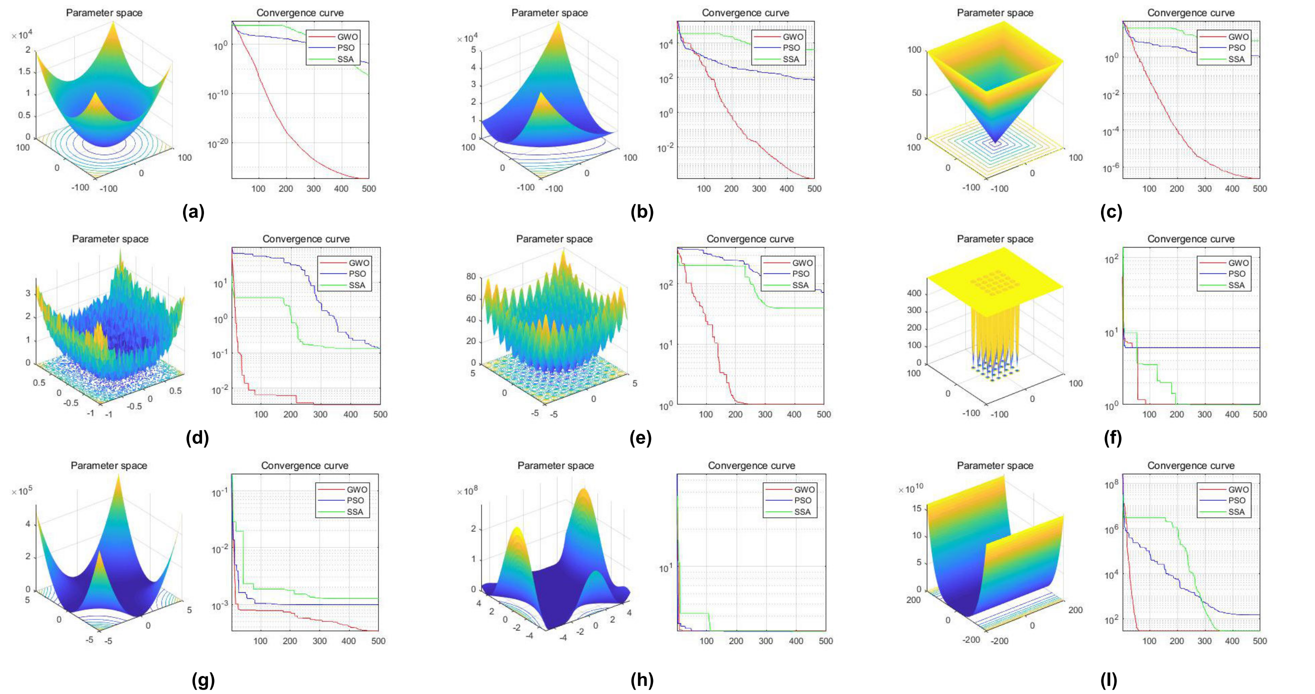

FIGURE 2. Optimization results of the function. The left figure in each pair shows the parameter space, and the right figure shows the convergence curve.

In an actual situation, the position of the prey is unknown, so we choose the best knowledge of the prey (obtained by an α, β, or γ wolf) to update the positions of all the other wolves (ω wolves) using Formulas (11) and (12). We regard the solution space of the fitness function as the range of the wolves. Since the T matrix has been obtained by Formula (4), we can conclude that the solution space is the value range of the three rotation angles, according to Formulas (3) and (1). Three group optimal angles of the fitness function are saved as the positions of α, β and γ wolves, and the remaining feasible solutions represent the positions of ω wolves. The positions of ω wolves are updated according to the positions of α, β and γ wolves, and the specific update method is as follows:

$$\begin{aligned} \vec{D}_\alpha &= \vec{C}_1^\circ \vec{X}_\alpha(t) - \vec{X}(t) \\ \vec{D}_\beta &= \vec{C}_2^\circ \vec{X}_\beta(t) - \vec{X}(t) \\ \vec{D}_\gamma &= \vec{C}_3^\circ \vec{X}_\gamma(t) - \vec{X}(t). \end{aligned} \tag{11}$$

$\vec{D}_\alpha$, $\vec{D}_\beta$ and $\vec{D}_\gamma$ represent the distances between α, β and γ and the current position of ω, respectively, and $\vec{X}_\alpha$(t), $\vec{X}_\beta$(t), and $\vec{X}_\gamma$(t) are the positions of α, β, and γ.

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha(t) - \vec{A}_1^\circ \vec{D}_\alpha \\ \vec{X}_2 &= \vec{X}_\beta(t) - \vec{A}_2^\circ \vec{D}_\beta \\ \vec{X}_3 &= \vec{X}_\gamma(t) - \vec{A}_3^\circ \vec{D}_\gamma \\ \vec{X}(t+1) &= \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \end{aligned} \tag{12}$$

$\vec{X}_1$, $\vec{X}_2$, and $\vec{X}_3$ are the effects on the location of ω due to α, β, and γ, respectively. $\vec{X}$(t + 1) is the updated position of ω.

In the process of capture, the convergence factor and parameter A gradually decrease. When $|\vec{A}| > 1$, the wolves are searching within the region; when $|\vec{A}| < 1$, the wolves launch an attack. That is, the convergence factor controls the search capability of the wolves. $\vec{C}$ is a random vector that contains random values in [0, 2], representing the random interference of the prey position. $|\vec{C}| > 1$ indicates that random interference is large, which leads to a strong stochastic disturbance and guarantees a higher possibility of outputting a locally optimal solution.

3) MATHEMATICAL PROCEDURE OF THE ALGORITHM

The main procedure of the GWO algorithm is as follows:

1) Initialize the wolf population by initializing multiple groups of rotation angles randomly;
2) Initialize the parameter vectors $\vec{\mathrm{A}}$ and $\vec{\mathrm{C}}$;
3) Obtain the fitness functions of all the wolves;
4) Select three solutions with the best fitness as the α, β and γ wolves;
5) Update the positions of all ω wolves and $\vec{\mathrm{A}}$ and $\vec{\mathrm{C}}$;
6) Stop if the iteration threshold is reached; otherwise, return to step (3).

V. EXPERIMENTAL RESULTS

To evaluate the performance of the PCR-GW algorithm, the bunny, dragon, and armadillo models of the Stanford University Computer Graphics Laboratory [2] are selected. Our method is compared with 4PCS, the algorithm based on FPFH, and the method based on the PSO algorithm.

TABLE 1. **The pseudocode of PCR-GW.**

Notation:
P_{key}, Q_{key}: key points of the P and Q point clouds
N: the population of grey wolves
R_i: a group rotation angle parameter in the R matrix
t, t_{iter}: iterations and iteration threshold

Input: source point cloud P, target point cloud Q
Output: registered point cloud

Solve the matrix T with the centralization method
obtain the central coordinates $\overline{\boldsymbol{P}}, \overline{\boldsymbol{Q}}$ from P and Q
$\boldsymbol{T} \leftarrow \overline{\boldsymbol{Q}} - \overline{\boldsymbol{P}}$
Extract key points from P and Q
for all points $p_i \in$ P **do**
    if p_i is a key point **then**
        $P_{key} \leftarrow p_i$
    end if
end for
determine Q_{key} in the same way
Use GWO to obtain the parameters of the R matrix
initialize the grey wolf population R_i (i = 1, 2, …, N)
initialize a, $\vec{\mathbf{A}}$, and $\vec{\mathbf{C}}$
$\vec{\mathbf{X}}_\alpha \leftarrow$ the parameter values with the best optimal fitness
$\vec{\mathbf{X}}_\beta \leftarrow$ the parameter values with the second best fitness
$\vec{\mathbf{X}}_\gamma \leftarrow$ the parameter values with the third best fitness
$t \leftarrow 0$
while $t < t_{iter}$ **do**
    for $i \leftarrow 1$ to N **do**
        obtain matrix R from R_i with Formula (1)
        calculate the fitness function of all wolves by Formula (3)
    end for
    update $\vec{\mathbf{X}}_\alpha$, $\vec{\mathbf{X}}_\beta$ and $\vec{\mathbf{X}}_\gamma$
    for $j \leftarrow 1$ to N **do**
        update the position of the current ω wolves by Formula (11) and Formula (12)
        update a, $\vec{\mathbf{A}}$, and $\vec{\mathbf{C}}$
    end for
    $t \leftarrow t + 1$
end while
put $\vec{\mathbf{X}}_\alpha$ into Formula (1) to obtain matrix R
use matrix R and T to register the point clouds

During the experiment, we defined a performance index to evaluate each algorithm, which includes the running time and registration error. The operating environment of the experiment includes an Intel Core-i5 CPU with 6 GB memory and a Windows 10 operating system. The algorithm proposed and its contrasted algorithms are all applied with MATLAB. MATLAB 2017b is used for compiling and registering the point cloud models.

A. PARAMETER SETTINGS

In our algorithm, some parameters need to be defined in the key point selection module and the parameter solution module. The values of these parameters should be set reasonably because they play an important role in the performance of the algorithm. In the key point selection module, we conducted several experiments based on the information in previous research and the actual situation. The appropriate parameters are set as follows: the number of neighborhood points $K = 6$, the neighborhood radius $r = 0.005$, $\varepsilon_1 = 0.7$, and $\varepsilon_2 = 0.4$.

After the key points are selected, the GWO algorithm is applied to solve the parameters of the rotation matrix. The population size and number of iterations in the algorithm exert a direct influence on the algorithm performance. Overly large or small populations can lead to a low operating efficiency for the algorithm. In addition, a small number of iterations will result in the early maturity of the algorithm. Therefore, we designed a set of experiments to find the most suitable parameters.

We used the Happy Buddha model that appears at the beginning of this article, taking the population size and the number of iterations as the variables in conducting the experiment. The root-mean-square error (RMSE) is used as the measurement standard and is given as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_p}\sum_{i=1}^{N_p}\left\| R * p_i + T - q_i \right\|}. \tag{13}$$

N_p is the number of key points chosen from P. RMSE is the mean distance of the corresponding points between two

TABLE 2. RMSE values of parameters (mm).

Parameters	5	10	15	20	25	30	35
100	6.9735	4.8724	3.5854	2.5958	2.1562	1.8082	1.6144
200	4.4375	3.1648	2.8794	1.9641	1.7597	1.5983	1.4154
300	3.8964	2.4684	1.8536	1.3595	1.3865	1.4561	1.3565
400	2.5763	2.0864	1.6854	1.3265	1.3436	1.3765	1.3597
500	2.0764	1.8685	1.5974	1.3558	1.4768	1.344	1.3250

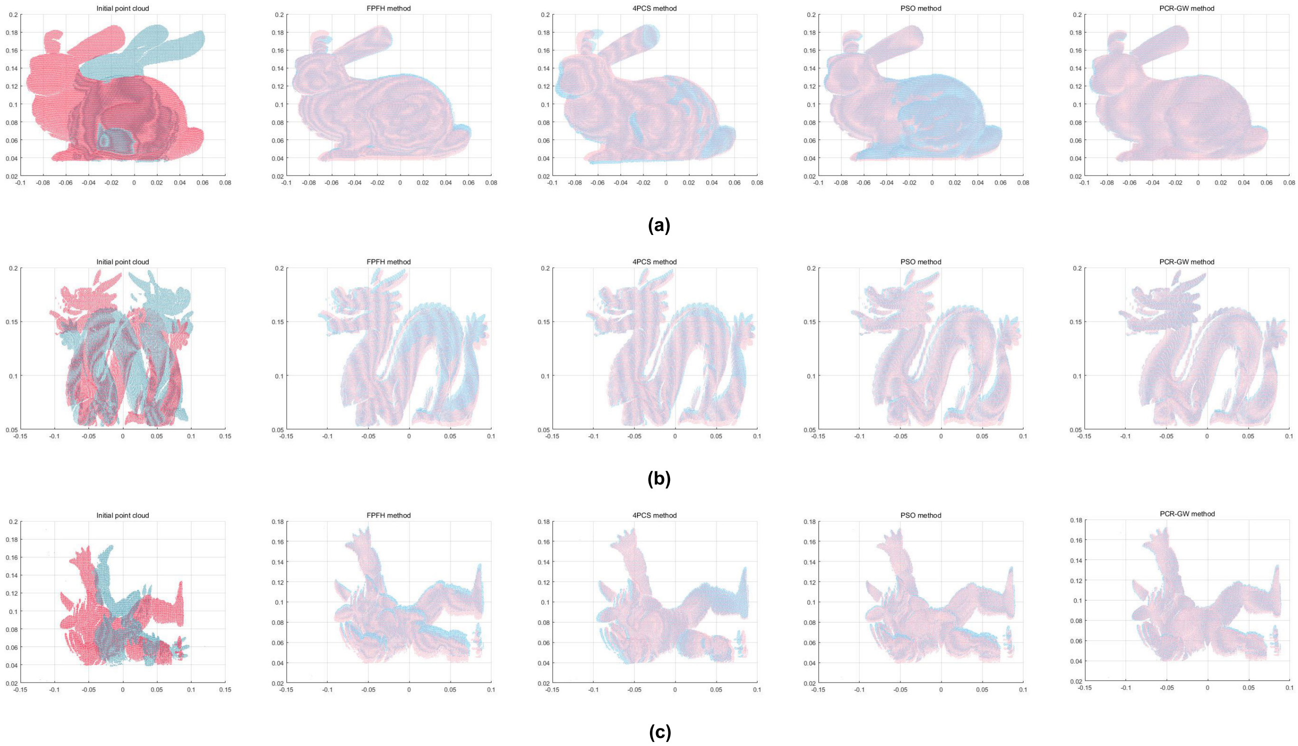

FIGURE 3. Registration results of the algorithms. From left to right are the original model, FPFH algorithm, 4PCS algorithm, PSO algorithm, and PCR-GW algorithm. (a) is the bunny model. (b) is the dragon model. (c) is the armadillo.

point clouds, which is a classic method of point cloud registration. As the registration effect is the most concerning part, a parameter value corresponding to a small RMSE is preferred. Table 2 shows the RMSE values of the different parameters. The horizontal parameters in the table correspond to the population size N, and the vertical parameters correspond to the number of iterations t.

According to Table 2, the final result remained at 1.3 mm and increased and decreased as the parameters changed. Considering that excessively large numbers of iterations and population sizes will slow down the computational speed, we should select the minimal parameters in the range of stable results. The information in Table 2 shows that when the population number $N = 20$ and the number of iterations $t = 300$, the accuracy can reach 1.3 mm. Although the accuracy is good when $N = 35$ and $t = 500$, considering that the accuracy changes relatively little but the time cost increases quickly, we still choose to set the initial population size to 20 and the number of iterations to 300.

B. RESULT ANALYSIS

Popular point cloud algorithms, including the registration algorithm based on FPFH features, 4PCS, and the PSO algorithm for parameter optimization, are selected to compare with the PCR-GW algorithm. The results indicate that the PCR-GW method performs well for the data set. The registration results are shown in Figure 3.

1) REGISTRATION ACCURACY EVALUATION

To evaluate the performance of each algorithm, the RMSE is used. Each experiment is repeated five times, and then the average RMSE is calculated to ensure the accuracy of the experiment. The RMSE values of the various algorithms are shown in Table 3. By comparing the errors of the various algorithms, it can be seen that all the above algorithms can provide good initial positions. Furthermore, PCR-GW outperforms the other three algorithms, and it provides a better position for fine registration. The registration of the PSO algorithm is

TABLE 3. The RMSE of the algorithms (mm).

Method	FPFH	4PCS	PSO	PCR-GW
bunny	3.1369	2.6754	1.9945	1.5937
dragon	4.3367	3.0945	1.8957	1.3074
armadillo	3.4956	2.5537	2.0253	1.7483

TABLE 4. The runtime of the algorithms (s).

Method	FPFH	4PCS	PSO	PCR-GW
bunny	35.48	26.74	49.67	33.35
dragon	56.82	47.98	61.73	54.88
armadillo	21.38	16.81	30.46	24.41

also good, while the registration position error of the 4PCS algorithm is relatively large.

2) TIME PERFORMANCE ANALYSIS

The runtime is an important index to measure the performance of algorithms. The average runtimes of the four algorithms are shown in Table 4. The runtimes of the algorithms show that PCR-GW takes less time than the same type of PSO algorithm. Compared with 4PCS, PCR-GW takes slightly more time but has more advantages given its accuracy. Due to the iterative process, our algorithm is not outstanding among all algorithms in terms of time; we will address this in future research.

VI. CONCLUSION

In this article, a point cloud registration algorithm based on the grey wolf optimizer (GWO) is proposed, and the key points are selected according to the ISS feature to simplify the point cloud. After that, the parameters in the transformation matrix are solved by the GWO algorithm to obtain the registration results. In this article, multiple experiments are conducted according to theoretical and practical experience to find the most appropriate parameters. The GWO algorithm effectively balances the global and local optimization abilities and can obtain the optimal value within a short time. Through comparison with other algorithms, it is shown that PCR-GW can obtain accurate registration results and ensure operational efficiency. In the future, we will focus on decreasing the runtime of the iteration process. (1) We can add machine learning to the process of finding the corresponding points (q_i and p_i). (2) We plan to optimize the method that finds the α, β, and γ wolf in each iteration. (3) We will consider using a more efficient fitness function.

REFERENCES

[1] Y. Díez, F. Roure, X. Lladó, and J. Salvi, "A qualitative review on 3D coarse registration methods," *ACM Comput. Surv.*, vol. 47, no. 3, pp. 1–36, Apr. 2015.

[2] *The Stanford 3D Scanning Repository*. Accessed: Aug. 19, 2014. [Online]. Available: http://graphics.stanford.edu/data/3Dscanrep/

[3] L. Cheng, S. Chen, X. Liu, H. Xu, Y. Wu, M. Li, and Y. Chen, "Registration of laser scanning point clouds: A review," *Sensors*, vol. 18, no. 5, p. 1641, 2018.

[4] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *Proc. SPIE*, vol. 16, no. 11, pp. 586–606, 1992.

[5] H. Sobreira, C. M. Costa, I. Sousa, L. Rocha, J. Lima, P. C. M. A. Farias, P. Costa, and A. P. Moreira, "Map-matching algorithms for robot self-localization: A comparison between perfect match, iterative closest point and normal distributions transform," *J. Intell. Robot. Syst.*, vol. 93, nos. 3–4, pp. 533–546, Mar. 2019.

[6] R. Marani, V. Renò, M. Nitti, T. D'Orazio, and E. Stella, "A modified iterative closest point algorithm for 3D point cloud registration," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 31, no. 7, pp. 515–534, Jul. 2016.

[7] X. Shi, J. Peng, J. Li, P. Yan, and H. Gong, "The iterative closest point registration algorithm based on the normal distribution transformation," *Procedia Comput. Sci.*, vol. 147, pp. 181–190, Jan. 2019.

[8] A. L. Pavlov, G. W. Ovchinnikov, D. Y. Derbyshev, D. Tsetserukou, and I. V. Oseledets, "AA-ICP: Iterative closest point with anderson acceleration," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 1–6.

[9] P. Kamencay, M. Sinko, R. Hudec, M. Benco, and R. Radil, "Improved feature point algorithm for 3D point cloud registration," in *Proc. 42nd Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2019, pp. 517–520.

[10] J. Yang, K. Xian, P. Wang, and Y. Zhang, "A performance evaluation of correspondence grouping methods for 3D rigid data matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Dec. 17, 2019, doi: 10.1109/TPAMI.2019.2960234.

[11] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2009, pp. 3212–3217.

[12] J. Yang, S. Quan, P. Wang, and Y. Zhang, "Evaluating local geometric feature representations for 3D rigid data matching," *IEEE Trans. Image Process.*, vol. 29, pp. 2522–2535, 2020.

[13] C.-S. Chen, P.-C. Chen, and C.-M. Hsu, "Three-dimensional object recognition and registration for robotic grasping systems using a modified viewpoint feature histogram," *Sensors*, vol. 16, no. 11, p. 1969, Nov. 2016.

[14] P. Cirujeda, Y. D. Cid, X. Mateo, and X. Binefa, "A 3D scene registration method via covariance descriptors and an evolutionary stable strategy game theory solver," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 306–329, Dec. 2015.

[15] Y. Ge, B. Wang, J. Nie, and B. Sun, "A point cloud registration method combining enhanced particle swarm optimization and iterative closest point method," in *Proc. Chin. Control Decis. Conf. (CCDC)*, May 2016, pp. 2810–2815.

[16] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," in *Proc. ACM SIGGRAPH Papers*, 2008, pp. 1–10.

[17] H. Hong and B. H. Lee, "Probabilistic normal distributions transform representation for accurate 3D point cloud registration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 3333–3338.

[18] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "DeepICP: An end-to-end deep neural network for 3D point cloud registration," 2019, *arXiv:1905.04153*. [Online]. Available: http://arxiv.org/abs/1905.04153

[19] G. K. L. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin, "Registration of 3D point clouds and meshes: A survey from rigid to nonrigid," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 7, pp. 1199–1217, Jul. 2013.

[20] D. Chetverikov, D. Svirko, and D. Stepanov, "The trimmed iterative closest point algorithm," in *Proc. Object Recognit. Supported User Interact. Service Robots*, no. 3, 2002, pp. 545–548.

[21] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[22] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, Jul. 2018.

[23] A. Jaafari, M. Panahi, B. T. Pham, H. Shahabi, D. T. Bui, F. Rezaie, and S. Lee, "Meta optimization of an adaptive neuro-fuzzy inference system with grey wolf optimizer and biogeography-based optimization algorithms for spatial prediction of landslide susceptibility," *Catena*, vol. 175, pp. 430–445, Apr. 2019.

[24] X. Ma, X. Mei, W. Wu, X. Wu, and B. Zeng, "A novel fractional time delayed grey model with grey wolf optimizer and its applications in forecasting the natural gas and coal consumption in Chongqing China," *Energy*, vol. 178, pp. 487–507, Jul. 2019.

[25] S. Gupta and K. Deep, "A novel random walk grey wolf optimizer," *Swarm Evol. Comput.*, vol. 44, pp. 101–112, Feb. 2019.

[26] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems," *J. Comput. Des. Eng.*, vol. 5, no. 4, pp. 458–472, Oct. 2018.

[27] Y. Xue, J. Jiang, B. Zhao, and T. Ma, "A self-adaptive artificial bee colony algorithm based on global best for global optimization," *Soft Comput.*, vol. 22, no. 9, pp. 2935–2952, May 2018.

YAQING FENG received the B.S. degree from Hunan Normal University, China, in 2018. She is currently pursuing the master's degree in optical engineering with the Beijing Institute of Technology. Her current research interests include 3D reconstruction and deep learning.

JIALIN TANG received the bachelor's and master's degrees in optical engineering from Sun Yat-sen University, Guangzhou, China, in 2006 and 2009, respectively. He is currently pursuing the Ph.D. degree in data science with the City University of Macau. He has worked as a Visiting Scholar with the South China University of Technology, from 2014 to 2015. He is currently a Senior Lecturer with the Key Laboratory of Photoelectric Imaging Technology and System, Ministry of Education of China, Beijing Institute of Technology, Zhuhai. His current research interests include computer vision, digital image processing, and artificial intelligence. He was awarded the Outstanding Teacher of Beijing Institute of Technology, in 2017.

BINGHUA SU received the Ph.D. degree in optical engineering from the Beijing Institute of Technology, Beijing, China, in 2002. He is currently the Dean of the Faculty of Information Technology, Beijing Institute of Technology, Zhuhai, and the Director of the Key Laboratory of Photoelectric Imaging Technology and System, Ministry of Education of China, Beijing Institute of Technology, Zhuhai. His research interests include optoelectronic imaging, computer vision, and computer graphics. He is a member of the Professional Committee of Optoelectronic Technology of China Optics Society. He is also the Executive Director of the Board of Optical Mechanical and Electrical Technology and Systems Integration Branch of China Instrumentation Society.

QINGLANG SU received the B.Sc. degree (Hons.) in electronics, in 1999, the Ph.D. degree in digital image processing from the University of Lancashire, U.K., in 2004, with a specialization on automatic image alignment for clinical evaluation of patient setup errors in radiotherapy, and the Ph.D. degree in applied psychology, in 2017. He is currently an Associate Professor and the Course Director of the Institute of Data Science, City University of Macau. He is also the Deputy Chief Supervisor of the Macao Software Industry Association and a Coordinator of research projects funded by different entities. His research interests include smart traffic, smart city, systems engineering, digital image processing, and big data processing and applications. He was awarded the full Scholarship for his Ph.D. degree in digital image processing.

ZHUANG ZHOU received the B.S. degree from the Xi'an University of Posts and Telecommunications, China, in 2017. He is currently pursuing the master's degree in optical engineering with the Beijing Institute of Technology. His current research interests include image dehazing, image enhancement, and underwater image restoration.